目 录

本产品由 PIC 单片机学习网制造,本产品完全兼容于 MICROCHIP 的 MPLAB-ICD2。

第一章 MCD2 如何工作

1.1 介绍

这一章对 PIC16-MCD2 如何工作提供了简单的描述. 它试图提供足够的信息，以便设计一块与 PIC16-MCD2 兼容的目标板，并能让 PIC16-MCD2对目标板进行编程与调试. 之所以描述编程和在线调试的基本原理,是为了方便在遇到问题时能快速的解决.

1.2 要点

这章提供了对 PIC16-MCD2 的总览，接着描述了 PIC16-MCD2 编程和调试模式. 这章对下列话题进行了讨论:

• MCD2 vs. ICE
• 模块接口连接
• 调试模式
• 针对调试模式的要求
• 调试保留资源
• 编程器模式

1.3 MCD2 vs. ICE

MCD2 在线调试器对于 ICE 在线仿真器是一个廉价的替代品. 它可以做很多以前需要在昂贵的硬件上才能实现的功能. 但这些好处是以在线仿真器的一些便利为代价的. 如果用户愿意将他们的应用设计成与 MCD2 兼容的，那他们将能享受到一个低价硬件调试器的好处.

与 ICE 相比，在线调试器要求:

• 在线调试器要求占用目标的一些硬件和软件资源.
• 目标PIC MCU必须有一个正常运行的时钟.
• MCD2只能在系统中的所有连接都完全正常时才能调试.

一个仿真器提供存储器和时钟，并能运行代码 － 即使没有与目标应

用板相连. 在开发和调试期间, ICE 提供了最强大的能力以获得系统所有的功能, 而 MCD2 如果在应用不能运行时将根本无法进行调试. 另一方面, 在系统量产之后, 仍然可以在应用板上放置一个在线调试连接座并与 MCD2 相连. 这样允许对应用方便的测试, 调试和再编程. 尽管 MCD2 与 ICE 相比有一些不足, 但它也有一些突出的优点:

• 在量产之后可直接与目标板相连, 而不需要先取下单片机再插上仿真头.

• MCD2可以对目标应用进行再编程, 而不需要其它连接或设备.

注意: ICE 使用传统的硬件来仿真目标微控制器. MCD2 使用目标微控制器上的硬件来做 ICE 的一些工作. MCD2 还通过在目标器件上运

1.4 模块化接口连接

MCD2 使用六芯的模块化接口电缆与目标 PIC MCU 相连. MCD2 连接器的引脚顺序按照目标 PIC 板显示如图 1-1.

注意: MCD2 电缆在各端是镜像连接的, 在 MCD2 模块上的连接与这里显示的连接是相反的.

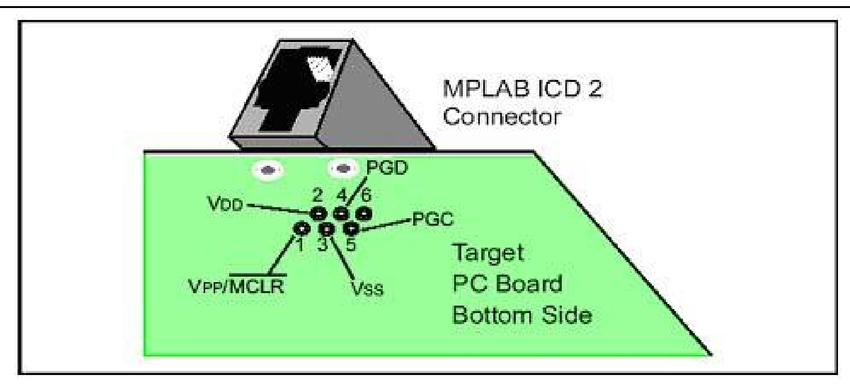
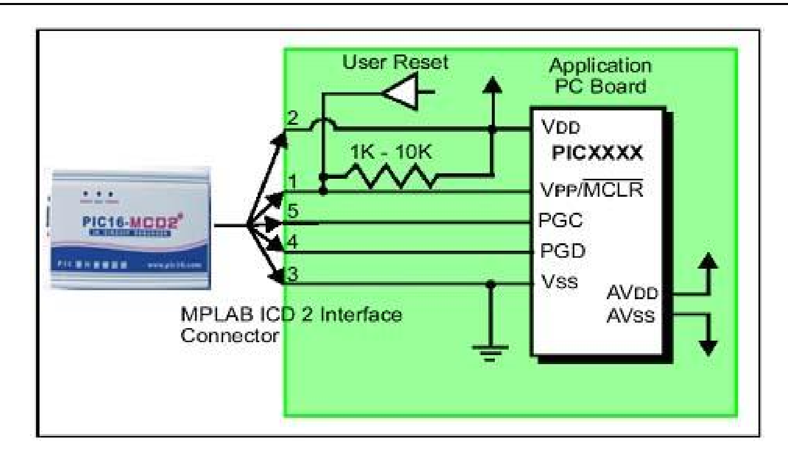


图 1-2 显示了 MCD2 与目标板上的模块化连接头的互连状况. 在 MCD2 连接头上有六个引脚, 但只使用了五个. 这个框图还显示了从连接头上的线与目标PIC板上的PIC MCU的连接情况. 在VPP/MCLR到VDD之间建议接一个上拉电阻(大约 10K 左右).



尽管在某些条件下引脚 2(VDD)能对目标应用供应有限的能量，但在这里引脚 2 和 3(VSS)将被忽略掉．在框图中它们被完整的显示出来，但在下面的描述中将只谈到与 MCD2 运行相关的三根线: VPP/MCLR, PGC 和 PGD.

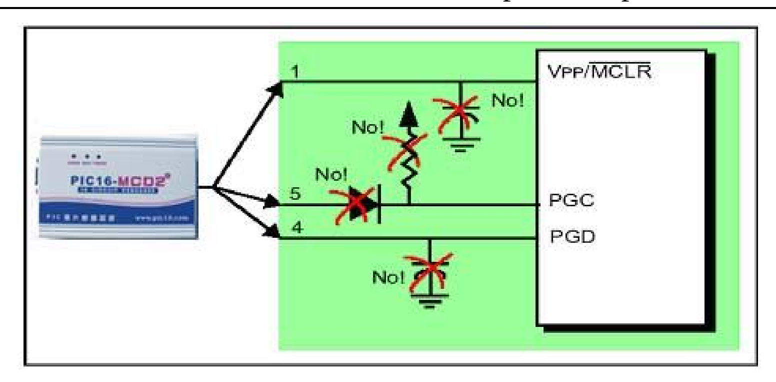
不是所有的 PIC MCU 都有 AVDD 和 AVSS，但如果它们出现在目标 PIC MCU 上，它们必须被连接上,以便 MCD2 能够运行.

> **注意:** 在下面的讨论中, VDD 被忽略掉了．但应注意到目标 VDD 也被用来对 MCD2 中的输出驱动器供电．如果 MCD2 在其 VDD 上没有电压，它将不能工作.

这个互连是很简单的．任何遇到的问题往往是由于在这几条线上其它的连接或器件对 MCD2 的运行造成干扰所造成的．具体讨论见 1.4.1 节.

1.4.1 将阻止 MCD2 工作的线路

下图显示了有效的 MCD2 连接加上一些器件将会妨碍 MCD2 的正常运行:



特别指出，这些线必须遵守下面规则:

• 在PGC/PGD上不要接上拉－它们将会分压，因为在MCD2中这些线有4.7K的下拉电阻.

• 在PGC/PGD上不要接电容－在编程和调试通讯期间，它们会阻止数据和时钟线上的快速转换.

• 在MCLR上不要接电容－它们会阻止VPP的快速转换．一个简单的上拉电阻通常就足够了.

• 在PGC/PGD上不要接二极管－它们会在MCD2和目标PIC MCU之间阻止双向通讯.

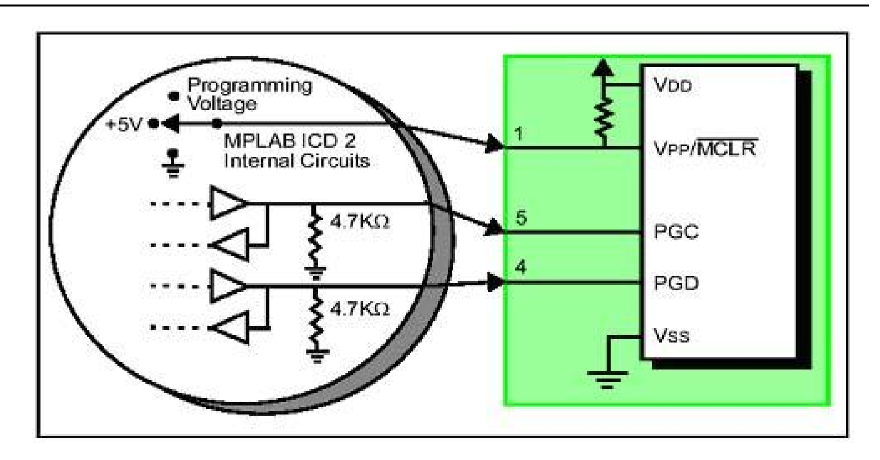
1.5 调试模式

使用 MCD2 作为一个调试器有两步．第一步要求将应用程序编程到目标 PIC MCU 中．第二步使用目标 FLASH PIC MCU 内部的在线调试硬件来运行和测试应用程序．这两步与 MPLAB IDE 的操作直接相关联:

1. 编程代码到目标.
2. 使用调试器设置断点和运行.

如果目标 PIC MCU 不能被正确的编程, MCD2 将不能进行调试.

图 1-4 显示了要求编程的基本互联状况．注意这个图和图 1-2 是一样的．但为了更简洁，来自 MCD2 的 VDD 和 VSS 没有显示.



MCD2 的部分内部接口电路的简化接线图如图所示. 对于编程时, 目标 PIC MCU 不需要时钟, 但必须要提供电源. 当编程时, MCD2 将编程电压加到 VPP 上, 在 PGC 上发送时钟脉冲, 并通过 PGD 发送串行数据. 要校验器件被正确的编程, 可以给 PGC 发送时钟,并通过 PGD 读回数据. 这符合了开发模式下 PIC MCU 的 ICSP 协议.

1.6 针对调试模式的要求

要使用 MCD2 进行调试(设置断点, 观察寄存器,等), 有几个关键的因素必须工作正确:

• MCD2必须与PC机相连. 它必须由一个外部电源. 并且它必须通过RS-232或USB与MPLAB IDE软件建立通讯.

• MCD2必须通过模块化接口电缆(或与之等效的)按照图示连接到目标PIC MCU的VPP, PGC和PGD引脚上. MCD2和目标PIC MCU之间也需要连接VSS和VDD.

• 目标PIC MCU必须有电源和工作正常的振荡器. 如果目标PIC MCU不工作,无论是什么原因,MCD2都不能调试.

• 目标PIC MCU必须将其配置字编程正确:
  - 振荡器配置位应与RC, XT, HS等相对应, 依靠目标设计的不同.
  - 目标PIC MCU的看门狗定时器一定不能使能.
  - 目标器件的代码保护不能使能.

- 目标器件的表读保护不能使能.

1.6.1 进入调试模式的操作顺序

如果上面的条件都能满足，那么当MCD2被设置成当前调试器时,可以执行下面的操作:

• 当选择***Debugger>Program***时,应用代码就会通过ICSP协议被编程到PIC单片机的存储器中.

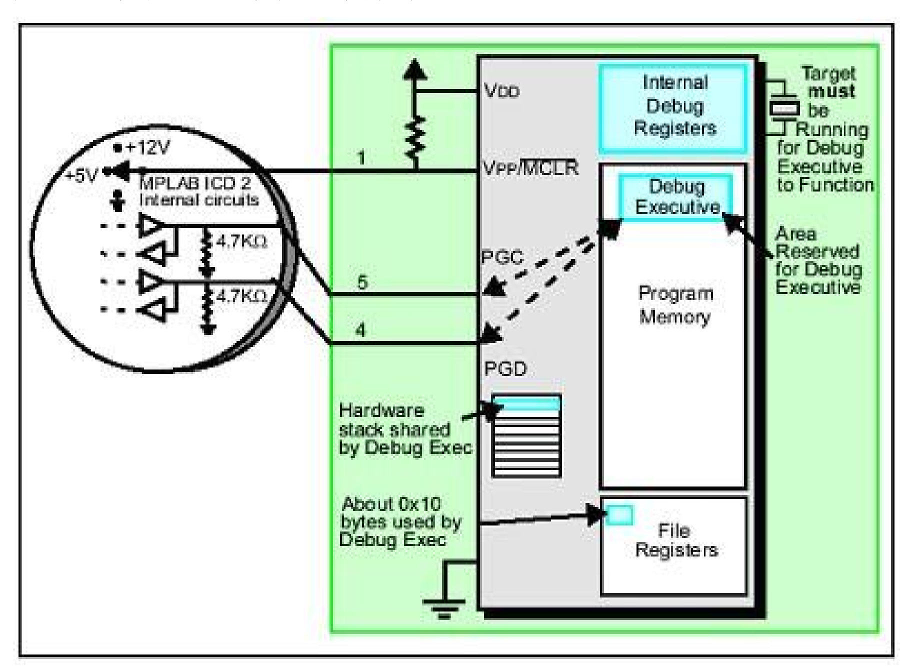
• 一小段"调试执行"程序被载入到目标PIC单片机的程序存储器高地址端. 由于这段调试代码必须占据在程序存储器中，所以目标程序就不能使用这段保留的空间. 这段调试代码通常需要程序存储器约0x120字.

• 目标PIC单片机的特殊的"在线调试"寄存器被使能. 这允许调试代码通过MCD2被激活.

• 通过将VPP/MCLR保持低,将使目标PIC单片机复位.

1.6.2 调试模式细节

图1-5显示了MCD2准备进行调试.



通常，为了发现一个应用程序是否能正确的运行，可以在程序代码中

先设置一个断点. 当断点从 MPLAB IDE 的用户接口被设置时, 断点的地址被保存在目标 PIC 单片机的特殊内部调试寄存器中.

下一步,按下 MPLAB IDE 的 ***Debugger>Run*** 功能或 ***Run*** 图标. MCD2 将抬升 VPP/MCLR 以允许目标运行. 目标将从地址零开始运行, 直到程序计数器达到前存储在内部调试寄存器中的断点地址.

在位于断点地址的那条指令被执行后, 目标 PIC 单片机的在线调试机构启动,并将 PIC 单片机的程序计数器传送给调试执行代码(很像一个中断), 用户的应用程序被有效的中止了. MCD2 通过 PGC 和 PGD 与调试代码进行通讯, 获得断点状态信息, 并将这些信息送回到 MPLAB IDE. 然后 MPLAB IDE 发送一系列的请求给 MCD2 以获得关于目标 PIC 单片机的信息, 如文件寄存器内容和 CPU 的状态. 这些请求最终通过调试代码来执行.

另一种获得断点的方式是按下 MPLAB IDE 的 ***"Halt"*** 按钮(在运行箭头右边的暂停标志). 这种方式将控制 PGC 和 PGD 线.这样目标 PIC 单片机的在线调试机构将程序计数器从程序存储器中的用户代码切换到调试代码. 再一次, 目标应用程序被有效的中止了. MPLAB IDE 使用 MCD2 和调试代码进行通讯以获得目标 PIC 单片机的状态.

1.7 调试保留资源

在使用MCD2进行开发时, 用户必须知道被保留用于调试的器件资源. 检查 MPLAB IDE 在线帮助或当前软件的 README 文件, 以找出处理器被使用的细节. 通常, 在线调试器使用了下面的片内资源:

- MCLR/VPP被共用作编程.
- 低电压编程(LVP)被关闭
- PGC和PGD被保留用作编程和在线调试.
- 一或二层堆栈被MCD2调试器使用
- 一些通用目标文件寄存器被保留.
- 程序存储器的高地址部分被保留用于调试执行代码.
- 在调试时PIC18FXXXX的影子堆栈不可用. 应用程序可以使用影子堆栈, 但不能从快速中断或快速CALL子程序返回, 因为MCD2在获得断点时将覆盖影子堆栈.

1.8 编程器模式

当使用 ***Programmer>Program*** 选项来编程器件时，在 MPLAB IDE 中在线调试寄存器应被关闭，这样 MCD2(MPLAB ICD2)将只编程目标应用代码和配置位(和 EEPROM 数据，如果有提供并选择)．调试执行代码将不被载入．在这种模式下，MCD2(MPLAB ICD2)只能控制 MCLR 线来复位和启动目标器件．断点不能被设置，寄存器的内容也不被查看．

MCD2(MPLAB ICD2)使用 ICSP 对目标进行编程．编程时不需要时钟，并且处理器的所有模式都能被编程，包括代码保护，看门狗定时器使能和表读保护．

> **注意:** 在使用 MCD2 调试极少引脚数的器件时会需要一个仿真头．这些器件按照前面描述的将 VPP,PGC 和 PGD 进行连接，不需要仿真头也能被编程．

第二章　初步介绍

2.1 介绍

这一章将描述 MPLAB IDE 软件和 MCD2(MPLAB ICD2)的使用，并将解释电源选型和推荐的上电顺序．第三章是使用 MCD2 一步一步的指南．在继续该指南之前，确信软件和硬件按照本章第 2.3 节和 2.5 节进行了配置．

2.2 要点

本章涵盖的话题包括:

- MCD2(MPLAB ICD2)系统组成
- 针对MCD2(MPLAB ICD2)来安装和配置MPLAB IDE
- 应用电源
- 总览：设置环境
- 总览: MCD2(MPLAB ICD2)开发过程

2.3 MCD2 系统组成

除了 MCD2 模块，还需要下面部件:

• MPLAB IDE软件(版本6.20或更高) － 安装在PC机上，用来控制MCD2.

• RS-232或USB电缆 － 连接MCD2到PC的串口或USB口.

> **注意:** 不要连接 USB 电缆，直到安装了 MPLAB IDE 软件并将设备的 USB 驱动配置完成.

• 模块化接口电缆 － 连接MCD2到演示板或用户应用板.

• 演示板或目标应用板 － 用来连接带片内调试能力的PIC单片机到模块化接口(和MCD2). 尽管从MPLAB IDE到MCD2的串行或USB通讯不需要目标连接也能建立，但MCD2不连接到目标板将不能作为一个调试器工作.

> **注意:** 在使用 USB 电缆时，在安装 MPLAB IDE 软件前不要和它连接.

• 电源适配器 － 对MCD2和目标板供电.

2.4 针对MCD2来安装和配置MPLAB IDE

要安装 MPLAB IDE 软件，首先要获得最新的 MPLAB IDE 安装程序(MP6xxxx.exe，其中 6xxxx 表示 MPLAB IDE 的版本). MPLAB IDE 可以从 Microchip 网站或配套的光盘中获得.

2.4.1 建立通讯

先安装好 MPLAB IDE，然后从 MPLAB IDE 下的 DriversXX 文件夹安装 USB 驱动. 查看第四章”USB 安装细节”. 如果使用 RS-232 电缆，将它连接到 MCD2 和 PC 机.

> **注意:** 不要将 USB 和 RS-232 都连接到 MCD2 上.

2.4.2 开始 MPLAB IDE

在安装完 MPLAB IDE 软件后，使用下面任何一种方法来启动它:

• 选择***开始>程序>Microchip MPLAB IDE>MPLAB IDE***.

• 双击***MPLAB IDE***图标

> **注意:** 关于使用MPLAB IDE软件的更多信息,可以查看:
> - "MPLAB v6.xx快速指南".
> - 针对MPLAB IDE的在线帮助
> - 包含在MPLAB IDE安装路径下的针对MPLAB IDE.txt的Readme文件.
> - 也可以查看关于MPLAB ICD2最新的Readme文件.

• 执行MPLAB IDE安装路径下\dlls子目录下的**mplab.exe**.

在启动MPLAB IDE后，它需要为MCD2(MPLAB ICD2)进行配置:

1. 选择MCD2(MPLAB ICD2)所支持的PIC器件.
2. 设置MCD2(MPLAB ICD2)作为当前调试器.
3. 配置MCD2(MPLAB ICD2)的RS-232或USB接口.

2.4.3 选择器件

使用器件选择对话框 ***Configure>Select Device*** 来选择MCD2(MPLAB ICD2)要调试的器件. MCD2(MPLAB ICD2)所支持的器件会在 Microchip 工具支持下面紧挨着 MPLAB ICD2 旁边有一个绿灯图标. 不支持的器件会有一个红灯图标. 黄灯图标表示那个器件可以选择，但它会有某些限制，可能不适合于量产工作.但它可以让 MCD2 做初步的使用.

2.4.4 设置 MPLAB ICD2 作为调试工具

选择 ***Debugger>Select Tool>MPLAB ICD2*** 将 MPLAB ICD2 作为调试工具. 一旦这个工具被选择后，调试菜单和 MPLAB IDE 工具条会显示调试选型. 同时，输出窗口会打开,关于 ICD 状态和通讯的信息会显示在 MPLAB ICD2 标签栏内.

> **注意:** MPLAB ICD2 既可以作为调试工具，也可以作为编程器. 但不要同时选择.

2.4.5 配置接口

跳到 ***Debugger>Settings*** 的通讯标签，选择 USB 接口或针对 RS-232 通讯正确的 COM 口.

2.5 应用电源

针对 MCD2(MPLAB ICD2)和目标板有许多的配置，下面是这些配置的关键:

• 当使用USB连接时, MCD2可以从PC机来供电．但必须要给目标板提供一个电源.

• 当使用RS-232连接到PC机时, MCD2必须接一个外部电源.

• 当MCD2有一个自己的电源时，它可以对较小的目标板在5V电压下提供一定量的电流,最高可达200mA.

• MCD2不能通过目标板供电.

• 在给目标板加电之前,应先给MCD2加电.

2.5.1 当MCD2给目标板供电时的上电顺序

使用下面的顺序来让MCD2给目标板供电．这个配置只允许在5V，最高200mA电流下使用.

1. 给MCD2加电．不要给目标板供电.
2. 启动MPLAB IDE6.xx.
3. 在MPLAB IDE的调试菜单下,选择**Connect**.
4. 在与MCD2建立通讯后，选择***Debugger>Settings***.
5. 在设置对话框中，点击Power标签，确认**"Power target circuit from MPLAB ICD2"**被选中．点击OK.

> **注意:** 如果Settings对话框Status标签栏的任何自测选型没通过，它将可能不能擦除和编程芯片．这将需要更多的故障检查．查看第五章和第六章或在线帮助以获得更多信息.

2.5.2 当目标板有一个独立电源时的上电顺序

使用这个顺序来给目标板通过它自己的电源来上电．对于这个配置，目标电源可以是 2-5V,电流可以比 MPLAB ICD2 提供的电流高．查看 PIC 单片机数据手册以确认这个器件所使用的操作电压范围.

1. 给MCD2加电．不要给目标板供电.
2. 启动MPLAB IDE6.xx.
3. 在MPLAB IDE的调试菜单下,选择**Connect**.
4. 在与MCD2建立通讯后，选择***Debugger>Settings***.

5. 在设置对话框中，点击Power标签，确认**"Power target circuit from MPLAB ICD2"**没有被选中．点击**OK**.
6. 给目标系统上电，然后选择***Debugger>Connect***.

> **注意:** 如果 Settings 对话框 Status 标签栏的任何自测选型没通过，它将可能不能擦除和编程芯片．这将需要更多的故障检查．查看第五章和第六章或在线帮助以获得更多信息.

2.6 总览：设置环境

这一章剩下的部分提供了对 MCD2 使用的总览，对一些功能和菜单进行了通用的描述．这一节提供了对 MCD2 操作的快速总结.

2.6.1 设置调试和编程选项

各种对话框允许用户来设置调制和编程选项:

• 配置位对话框(***Configure>Configuration Bits***)－在PIC处理器上选择配置位．对于这些选项完整的细节，可以查看这个器件数据手册特殊特性－配置位章节.

> **注意:** 配置位可以在源代码中被指定，以替代在这个对话框中设置．当这样做时，每次重建工程，配置位都将设置成源代码中所指定的值.

• 编程标签, ICD设置对话框(***Debugger>Settings***或***Programmer>Settings***)－设置编程选项(选择存储器，程序和外部存储器范围, ID，和编程选项，全部擦除)．如果期望编程ID位，可以在***Configure>ID Memory***对话框中设置要编程的值.

• 电源标签,ICD设置对话框(***Debugger>Settings***或***Programmer>Settings***)－检查MPLAB ICD2和目标电压值，如果MPLAB ICD2连接了一个电源(5V,最高200mA),选择"**Power the target from the MPLAB ICD2**".

• 状态标签，ICD 设置对话框(***Debugger>Settings*** 或***Programmer>Settings***)－除了在上电可以自动连接外，还可以在这个对话框中设置输出信息到一个文件．这些信息可以协助用户,或者如果可能，协助 Microchip 技术支持发现问题.

2.6.2 创建一个工程

最简单的创建一个新工程的方法是选 ***Project>Project Wizard***.在工程向导的帮助下，一个新的工程和用于建立工程的语言工具将被创建. 向导将指导用户添加源程序，库文件,链接文件等到工程窗口的各个节点. 第三章的指南会给出使用这个向导更多的细节.

2.6.3 建立工程

在创建工程之后，选择 ***Project>Build All*** 来建立应用. 这将创建这个应用的目标代码,用于 MCD2 给目标板编程.

2.7 总览: MCD2 开发过程

使用 MCD2 的开发过程包括四个步骤:

1. 使用 MPLAB ICD2 作为当前调试工具，将应用代码编程到目标应用中去.
2. 调试应用程序.
3. 修改源程序，重建工程. 然后重复上述工作，直到应用按照设计正常运行.
4. 选择 MPLAB ICD2 作为当前编程器，然后对目标器件编程.

2.7.1 作为调试器对目标处理器编程

作为调试器时，将应用代码编程到 MPLAB ICD2,可以按照下面步骤进行:

1. 选择 ***Debugger>Settings***，然后点击程序标签，为这个应用设置程序存储器的起始和结束地址.
2. 选择 ***Configure>Configuration Bits***，设置相应的配置位.
3. 选择 ***Debugger>Program***.

2.7.2 调试应用

要实时执行代码，可按照下面步骤:

1. 打开源程序(双击工程窗口的文件名或使用 ***File>Open***)或程序存储器窗口(***View>ProgramMemory***)用于观察.
2. 选择 ***Debugger>Run***(或点击工具栏 ***RUN*** 图标).处理器将一直运行直到碰到一个断点,或处理器通过选择 ***Debugger>Halt***(或点击工具栏 ***Halt*** 按钮)被中止.

   选择 ***Debugger>Breakpoints*** 为断点指定位置. 或在源程序的行或希望设置断点的程序存储器处点击右键，选择 ***Set Break Point***. MCD2

一次只能设置一个断点.

处理器被中止之后，代码可以以单步模式执行. 要执行单步，选择 ***Debugger>Step Into***(或点击工具栏 ***Step Into*** 按钮). 在每次单步之后处理器被中止.

> **注意:** 当执行单步时，中断被关闭.所以中断程序将无法进入. 可以在中断程序中设置断点，然后运行到那，再单步执行中断代码.

2.7.3 修改目标应用代码并重建 HEX 文件

要修改目标代码并重建 HEX 文件，可以按照下面步骤进行:

1. 打开源程序(双击工程窗口的文件名或使用 ***File>Open***).
2. 对调试代码做适当的修改.
3. 使用 ***Project>Build All*** 重建 HEX 文件.
4. 选择 ***Debugger>Program*** 将更新的 HEX 文件编程到目标器件.

2.7.4 完成应用: 使用 MCD2 作为编程器

一旦用户完成了代码调试,并且按照期望值运行了，那么器件就可以不需调试使能进行编程了. 这时用于操作的器件保留资源就可以使用了.

> **注意:** MCD2 既可以作为调试工具,又可以作为编程器. 但不要同时选择两者.

首先，从 ***Debugger>Select Tool*** 菜单设置调试器到”**None**”.

然后，选择 ***Programmer>Select Programmer>MPLAB ICD2*** 将 MCD2 设置为编程器.一旦这个工具被选择，编程器菜单和 MPLAB IDE 工具栏将会变更为显示编程器选项. 同时，输出窗口会被打开，在 MPLAB ICD2 栏上会显示与 MCD2 状态和通讯相关的信息.

现在应重建工程，以便所有的调试模式被关闭，并且在源程序中被定义的配置位将被编程到目标器件.

编程选项(即存储器区域和范围)可以在 ***Programmer>Settings*** 对话框的编程标签进行设置.

配置位将按照源文件的定义进行设置. 它们可以通过使用 ***Configuration>Configuration Bit*** 对话框手动选择.

如果希望的话，可以选择 ***Configure>ID Memory*** 来设置 ID 位.

第三章 MCD2 指南

3.1 介绍

这则指南将会使用例程 mcd-led1.asm 来介绍开发一个简单工程的过程. 这个例程在MCD2的产品配套光盘的MCD-DEMO编程实例文件夹中可以找到. 这个程序使用本站 MCD-DEMO 演示板来实现 PIC16F87X 的流水灯.

3.2 要点

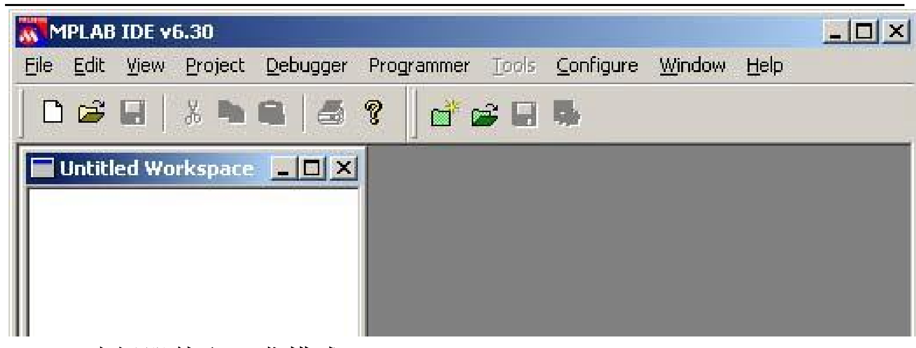
这章覆盖的话题包括:

- 设置环境
- 运行工程向导
- 查看工程
- 创建HEX文件
- 设置调试选型
- 设置演示板
- 载入调试代码
- 运行mcd-led1
- 调试mcd-led1
- 编程
- mcd-led1主程序和源代码

3.3 设置环境

在开始这章指南之前, 按照第二章第2.3节至2.5节的步骤设置硬件和MPLAB IDE 软件. 在这章指南中的一些初始设置可能在第二章中已经设置好.

一旦启动, MPLAB IDE 桌面应该显示如图 3-1.



3.3.1 选择器件和开发模式

选择器件:

1. 选择 ***Configure>Select Device***.
2. 在器件选择对话框，从器件列表框选择 PIC16F877A．在 Microchip 工具支持部分挨着 MPLAB ICD2 的灯图标应该为绿色.
3. 点击 **OK**.

选择 MCD2 作为调试器:

1. 选择 ***Debugger>Select Tool>MPLAB ICD2***．调试器菜单将会显示可供使用的其它调试选项．同时，输出窗口会打开显示连接信息.

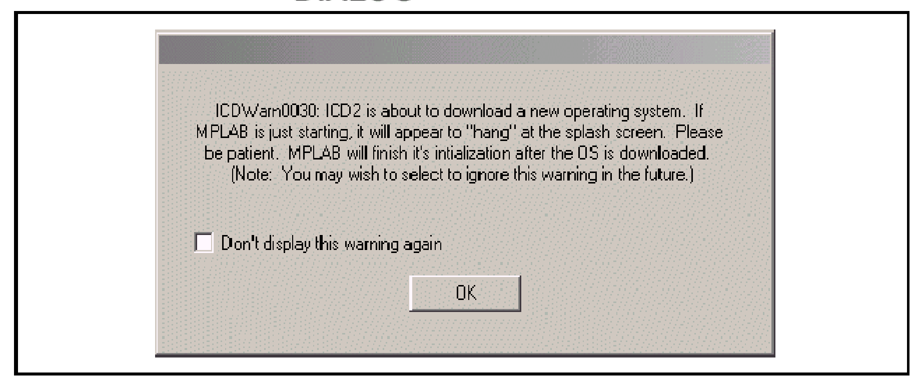
> **注意:** 如果 MPLAB IDE 在启动时试图连接到 MCD2,但因为 USB 口或一个非 COM1 的 COM 口被使用而失败，那么允许失败,继续第二步的通讯端口设置.

2. 选择 ***Debugger>Settings***，通讯标签．选择 USB 或 COM 端口/波特率.
3. 如果希望 MCD2 自动连接，点击状态标签，选择"**Automatically connect at startup**".检查信息级别设置为最小模式.
4. 点击 **OK**.
5. 选择 ***Debugger>Connect*** 连接到 MCD2.

3.3.2 更新 MCD2 固件(操作系统)

根据 MPLAB IDE 的版本或器件的选择，会提示一条信息显示固件需要更新. MPLAB IDE 将自动安装新的固件．更新对话框如图所示:

FIGURE 3-2: UPDATING MPLAB ICD 2 FIRMWARE DIALOG



同时，由于不同的 MPLAB ICD2 估计用于不同系列的 PIC 单片机，当切换不同的 PIC 单片机时也可能出现这个对话框.

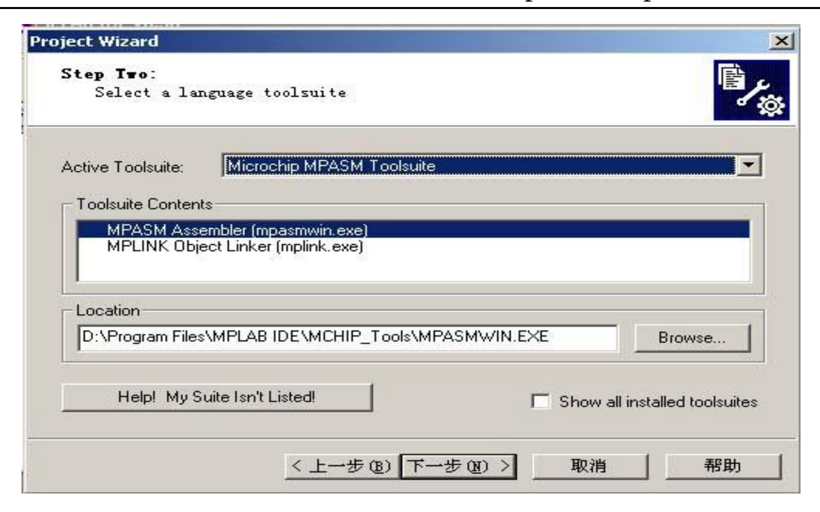
固件可以按照下面的步骤手动更改:

1. 选择 ***Debugger>Download ICD2 Operating System***. 选择 ICD2 固件文件对话框会打开.
2. 从列表中或浏览器中选择要下载的固件文件. 文件名的组成为 icdxxxxxx.hex,其中 xxxxxx 为版本号. 关于 MPLAB IDE 提供最新版本的信息可以查看 README 文件或 MPLAB ICD2.txt.根据选择的 PIC 器件版本可能会变动.
3. 点击 **Open**. MPLAB IDE 将下载新的操作系统到 MCD2.

3.4 运行工程向导

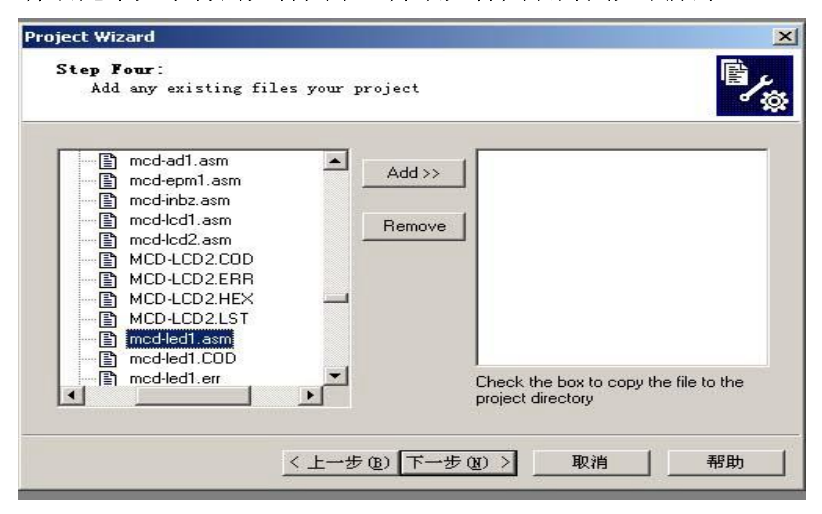
在这个工程中, MPASM 汇编器会被使用.

1. 选择 ***Project>Project Wizard*** 来建立第一个工程.
2. 进行到第二个向导对话框, PIC16F877A 应该被选中.
3. 继续进行到向导的下一个对话框，设置 MPASM 作为语言工具. 在 "**Active Toolsuite**" 下拉列表中选择 "**Microchip MPASM Toolsuite**". 确认 MPASM 和 MPLINK 设置在 **MPLAB IDE\MCHIP_Tools** 文件夹的相应可执行文件上. MPASM 应指定到 mpasmwin.exe,MPLINK 应指定到 mplink.exe.

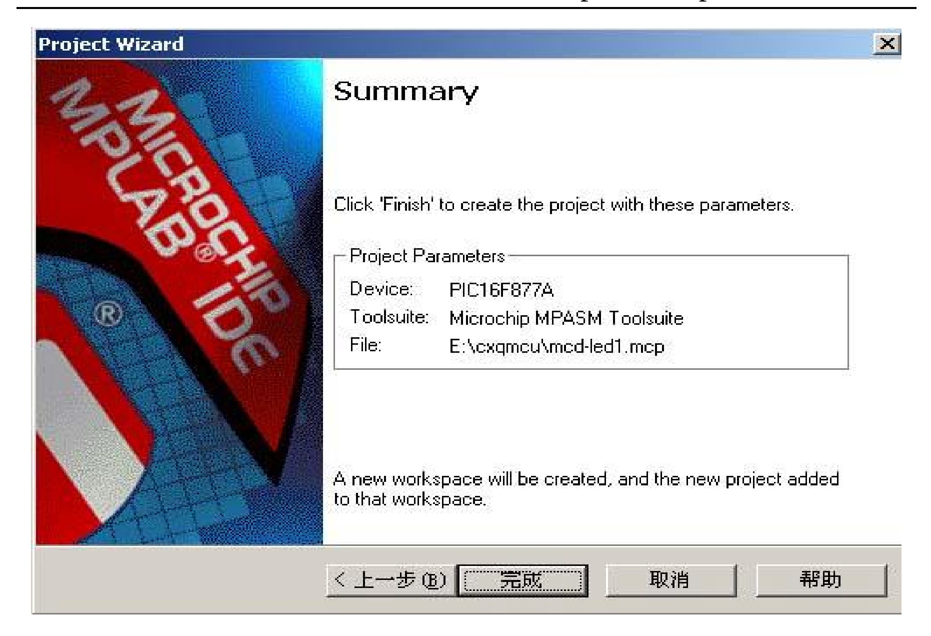


接着工程文件被添加到向导.

文件也可以在以后添加. 对于这个例子, 选择 MCD-LED1.ASM（请先将光盘的 MCD-DEMO 编程实例文件夹复制到本地硬盘根目录或其它上层文件名无中文字符的文件夹下，并改文件夹名为英文或数字.）



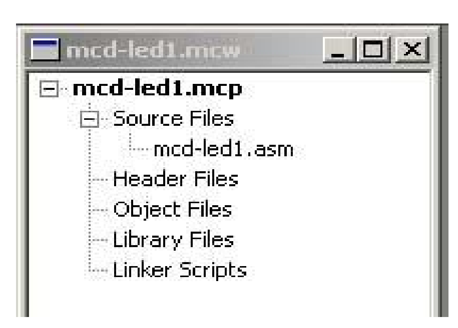
点击 MCD-LED1.ASM 来选中它，然后点击 **ADD>>**将它添加到右边的面板.



3.5 查看工程

退出向导后, MPLAB IDE 桌面又会出现. 关闭桌面其它的窗口以查看工程窗口.

工程窗口应该如下图所示.

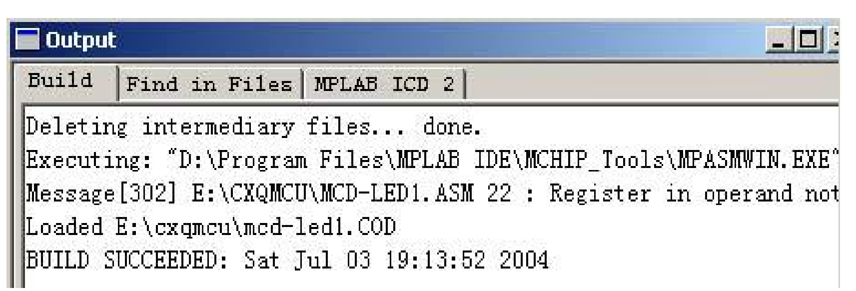


如果有错误, 文件可以手动的添加到工程窗口. 将光标移动到"**Source Files**"或"**Linker Scripts**"来添加文件. 要删除一个文件, 可以先选中它, 然后点击鼠标右键, 从菜单中删除文件.

3.6 创建 HEX 文件

要为调试创建一个 HEX 文件，选择 ***Project>Build All*** 或在工程窗口里的工程名字上点击右键并从弹出的菜单上选择"**Build All**". MPASM 汇编器总会以.asm 文件相同的名字生成一个.hex 文件.

输出窗口应该如图所示.



3.7 设置调试选项

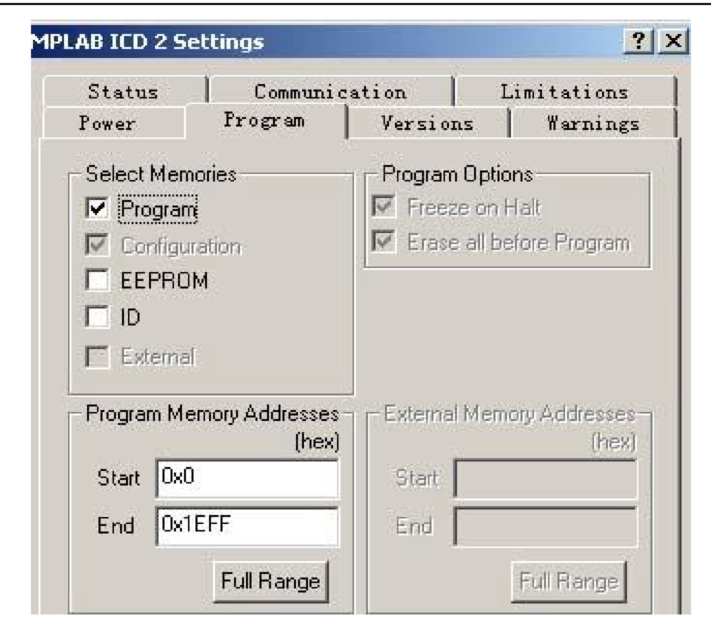
3.7.1 配置位

要设置配置位，选择 ***Configure>Configuration Bits***. 通过点击"**Settings**"列里的文本可以对它们进行修改. 在这个对话框中，针对这章指南,下面的配置位应该被设置:

- Oscillator – XT(这是针对MCD-DEMO. 如果使用其它的目标板，应进行相应的变更)
- Power Up Timer - Enabled
- Brown Out Detect - Disabled
- Watchdog Timer - Disabled
- Low Voltage Programming - Disabled
- Background Debug – Enabled(这将使能在线调试特性. 通常在选择MPLAB ICD2作为当前调试器时，这项会自动使能.)
- 其它所有配置位应被关闭.

3.7.2 编程选项

要设置编程选项，选择 ***Debugger>Settings*** 然后点击 **Program** 标签.



• 选择存储器部分应该是"**Program**"选中, "**EEPROM**"和"**ID**"不选. 当使用MCD2作为调试器时, 配置位将一直被编程, 配置选择框将被选中并为灰色.

• 程序存储器地址(起始地址和结束地址)用来设置程序存储器的范围, 它将被读, 编程或校验. 点击**Full Range**设置基于选中器件可供使用最大程序存储器的地址范围. 注意程序存储器的结束地址做了调整, 以保留空间给MCD2调试程序. 当使用MCD2作为调试器时, 程序不能超过上面的上限.

3.8 设置演示板

在开始调试前, 确认 MCD-DEMO 演示板按如下进行设置:

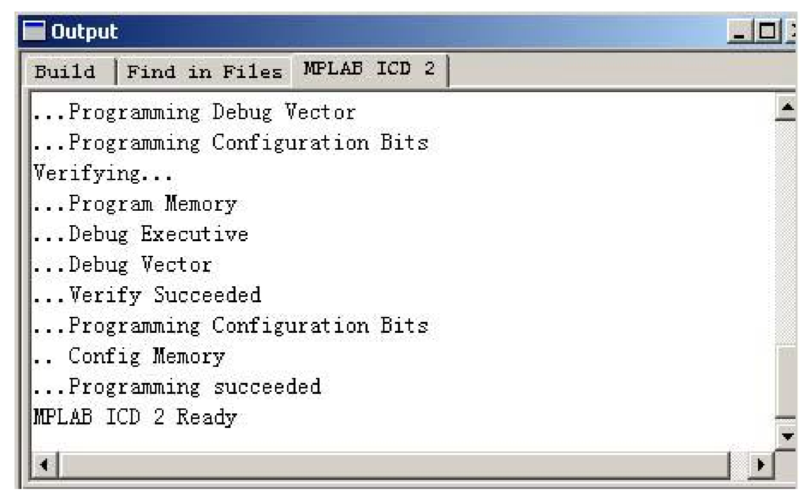
- 使用跳线J3,J4选择XT OSC选项
- 跳线S1全部ON.
- 电源被提供.

3.9 载入调试代码

选择 ***Debugger>Program*** 将 MCD-LED1.hex 编程到 MCD-DEMO 演示板上的 PIC16F877A.

注意: 调试代码会自动编程到程序存储器的顶层空间用于 MCD2 的调试. 调试代码必须被编程到目标 PIC 单片机, 以使用 MCD2 的在线调试能力.

编程可能需要几分钟. 在编程过程中, 输出对话框的 MPLAB ICD2 栏会显示操作的当前阶段. 当编程完成时, 对话框应该如图 3-10 所示.



3.10 运行 MCD-LED1

MCD2 可以以实时或单步模式执行.

- 在MPLAB IDE的运行模式中,MCD-DEMO演示板上的PIC16F877A将运行在实时模式.
- 在处理器被中止后,单步执行就能被访问.

工具栏按钮可以用于常用调试操作的快速访问.

调试菜单	工具栏按钮	调试菜单	工具栏按钮
运行		单步执行	
中止		复位	

开始实时模式:

1. 打开 MCD-LED1.asm 文件(在工程窗口上双击文件名或使用 File>Open).
2. 选择 ***Debugger>Run***(或点击工具栏 ***RUN*** 按钮)
3. 在演示板上，观察 LED 的流动过程,状态.
4. 选择 ***Debugger>Halt***(或点击工具栏 ***HALT*** 按钮)中止程序运行.
5. 选择 ***Debugger>Reset*** 复位程序.

3.11 调试 MCD-LED1

下面的任何一项都能阻止 MCD-LED1 程序的工作.

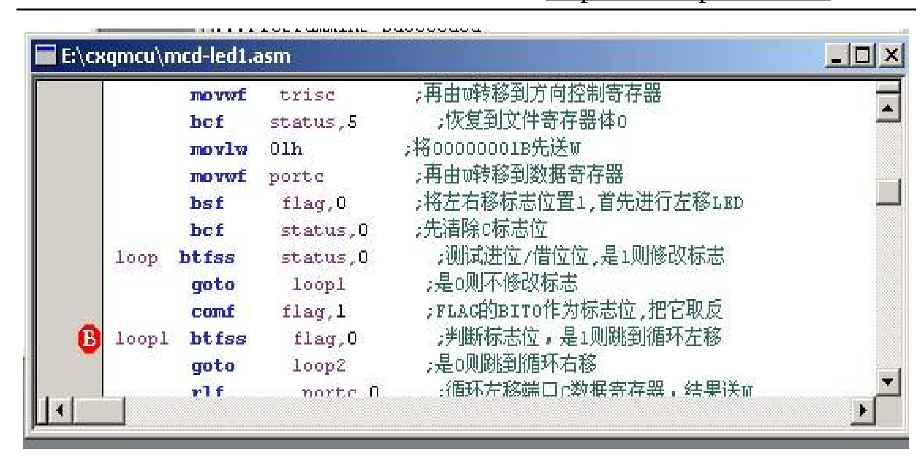
- DEMO开关未被设为ON.
- DEMO板的振荡方式跳线设置不符.
- 源程序中书写错误会导致程序工作不正常.

要观察寄存器的改变状态，可以在程序相关处设置断点.

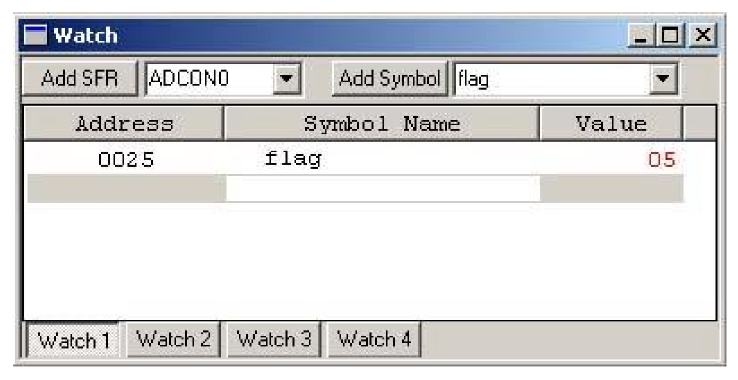
1. 从 MCD-LED1.asm 中选中或将光标放置在要设断点的下面一行如:
   btfss　　flag,0　　　;判断标志位，是1则跳到循环左移
2. 点击右键，显示弹出菜单.
3. 从弹出菜单选择 **Set Break Point**. 这一行现在被标记为断点(红色的 B 停止标记). 如图所示.
4. 选择 ***Debugger>Run***(或点击工具栏 ***RUN*** 按钮)以实时模式再次运行程序.

当程序执行到断点设置行时，断点会中止程序的运行. 然而，例程不会被中止.

5. 选择 ***Debugger>Halt***(或点击工具栏上 ***HALT*** 按钮)来中止程序.



6. 要复位程序，选择 ***Debugger>Reset***. 第一条指令应该显示带一个绿色的箭头.
7. 打开一个新的观察窗口来观察程序执行时 FLAG 寄存器值的变化. 选择 ***View>Watch***. 观察对话框将 **Watch_1** 选中打开. 从添加 Symbol 旁边的列表中选择"**flag**"，然后点击按钮"**Add Symbol**".**flag** 被添加到观察窗口中..如图所示.



8. 单步执行程序，选择 ***Debugger>Step***(或点击工具栏 ***STEP*** 按钮).
9. 在观察窗口中检查寄存器 FLAG 的值.
10. 选择***File>Save***保存文件.
11. 选择***Project>Build ALL***重建工程. 一条信息会提示程序被重新建立. 要使改变生效, MCD2必须被重新编程.

12. 选择***Debugger>Program***重新编程. 当MPLAB ICD2对话框显示'…Programming succeeded',程序就可以再次进入运行模式.
13. 在之前设置断点的行点击右键, 选择***Remove Break Point***.
14. 选择***Debugger>Run***使程序运行在实时模式. 旋转电位器来改变在LED上显示的值.

这则指南中的源程序只包含一个错误. 然而, 实际的代码可能会有更多问题. 使用MCD2和MPLAB IDE调试功能, 用户能成功的发现并修复这些问题.

3.12 编程

当程序成功的调试并运行时, 通常下一步就是对 PIC 单片机编程. 当进行这一步时, 被 MCD2 保留的资源将释放给用户.

对应用进行编程, 可以按照下面步骤:

1. 选择 ***Debugger>Select Tool>none*** 关闭 MPLAB ICD2 作为调试工具.
2. 在 ***Programmer>Select*** 菜单中选择 MPLAB ICD2 作为编程器.
3. 在 ***Programming>Settings*** 程序标签上设置用于编程的参数.
4. 选择 ***Programmer>Program***.

现在 MCD2 就能复位并运行目标(MCD2 也可以从目标断开, 在选择复位按钮之后, 应用会运行).

第四章 USB 安装细节

4.1 介绍

这一章介绍了针对不同 Windows 操作系统 USB 驱动安装的细节. 这些系统包括:

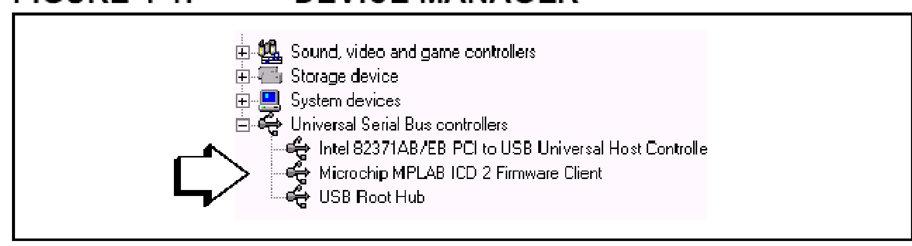
- Windows 98/ME
- Windows 2000
- Windows XP

如果安装了错误的驱动, 将会遇到USB问题. Windows经常会试图安装默认的USB驱动. 这个驱动将不能在MCD2上工作. 如果遇到了问题, 可以查看硬件设备管理器来检查Microchip USB驱动被列在USB设备下.

有时一个未知的USB设备会被列出在其它设备下．如果发现这个设备，应先将它删除．然后根据相应的操作系统继续下面的步骤．

图4-1显示了正确的配置:

FIGURE 4-1: DEVICE MANAGER



4.2 WINDOWS 98 和 WINDOWS ME 安装

4.2.1 步骤 1

当第一次将一个新的 USB 设备接到 Windows 98/ME 系统时，一个对话框会出现显示系统发现了一个新的硬件设备．也可能出现一对话框告诉用户设备数据库正在被建立或更新．

这些对话框出现之后，会显示”添加新硬件向导”对话框．点击下一步继续安装．

4.2.2 步骤 2

确认” 搜寻最合适的驱动…”被选中，然后点击下一步．

> **注意:** 如果在下面的步骤中没有发现或识别 INF 或设备驱动文件，重复步骤 2，选择”显示所有驱动列表…”．下一屏应该询问设备类型．这时点击磁盘安装按钮．接着的对话框允许选择可以找到.INF 文件的路径．一旦这个路径被选择，点击 **OK** 继续安装．

4.2.3 步骤 3

这个对话框允许用户指定哪里可以找到新设备所需要的文件．确认所有的选框都未选中，除了提示”指定位置”的那一个．选择”指定位置”，任何进入设备驱动文件的路径(可以键入路径，或使用浏览按钮进行搜索)．在

普通安装下，这个路径应该是:

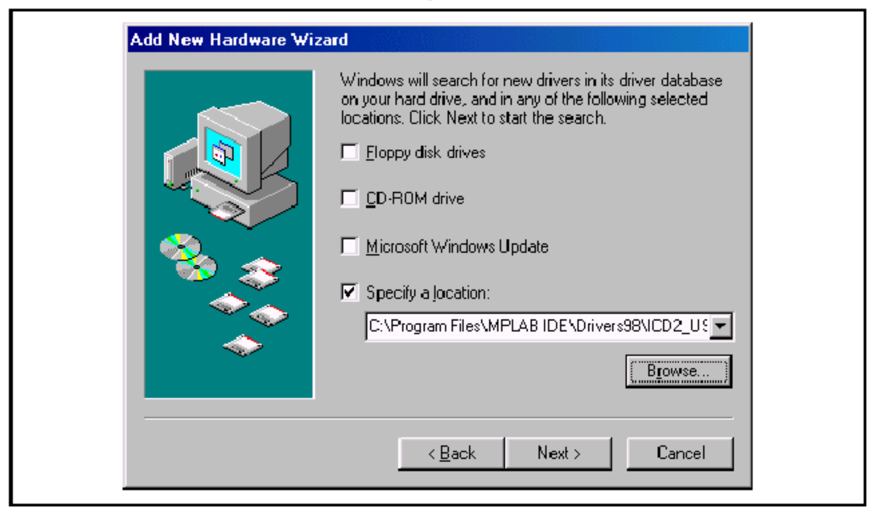
C:\Program Files\MPLAB IDE\Drivers98\ICD2_USB

这个路径必须包含下面的文件:

- ICD2W98.INF
- ICD2W98.SYS
- ICD2W98L.SYS

一旦路径被指定，点击下一步.

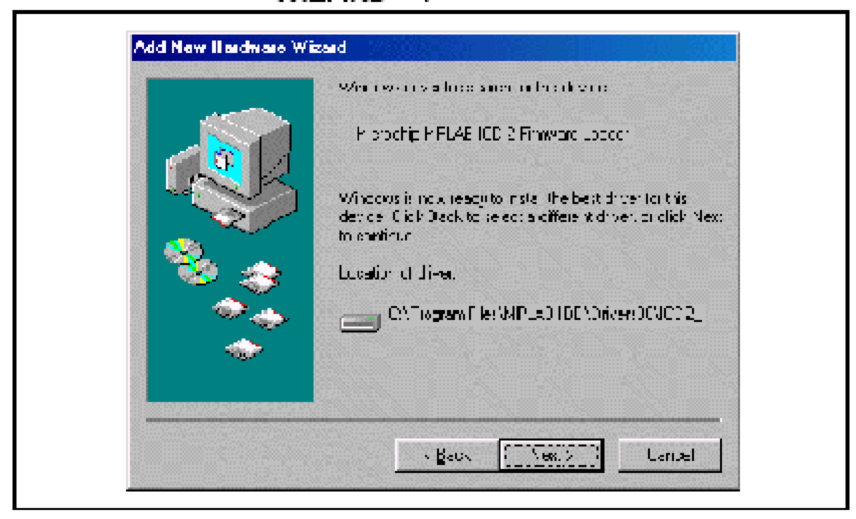
FIGURE 4-4: WINDOWS 98/ME NEW HARDWARE WIZARD - 3



4.2.4 步骤 4

下一个对话框应该显示”**Microchip MPLAB ICD2 Firmware Loader**”被找到，并显示之前被选择的路径. 点击下一步.

FIGURE 4-5: WINDOWS 98/ME NEW HARDWARE WIZARD - 4

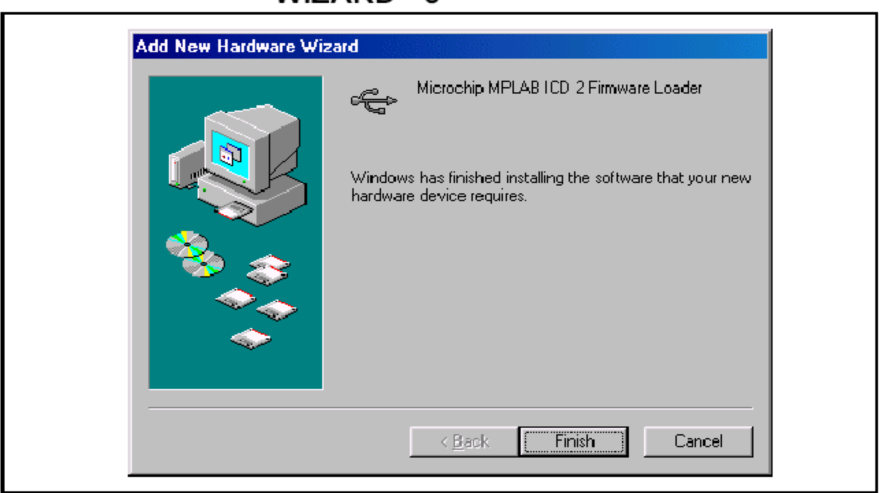


4.2.5 步骤 5

最后一个对话框会显示”**Microchip MPLAB ICD2 Firmware Loader**”被安装．点击完成结束安装.

这时 MPLAB ICD2 会再次被检测到．系统应该自动载入 MPLAB ICD2 USB 客户端驱动．在新的对话框请求另一个路径时，请指定与第三步相同的路径.

FIGURE 4-6: WINDOWS 98/ME NEW HARDWARE WIZARD - 5



4.3 WINDOWS 2000 安装

当 MPLAB ICD2 在 WINDOWS 2000 系统上第一次插入 USB 时，下面的步骤将被进行，以便正确的安装设备驱动.

4.3.1 步骤 1

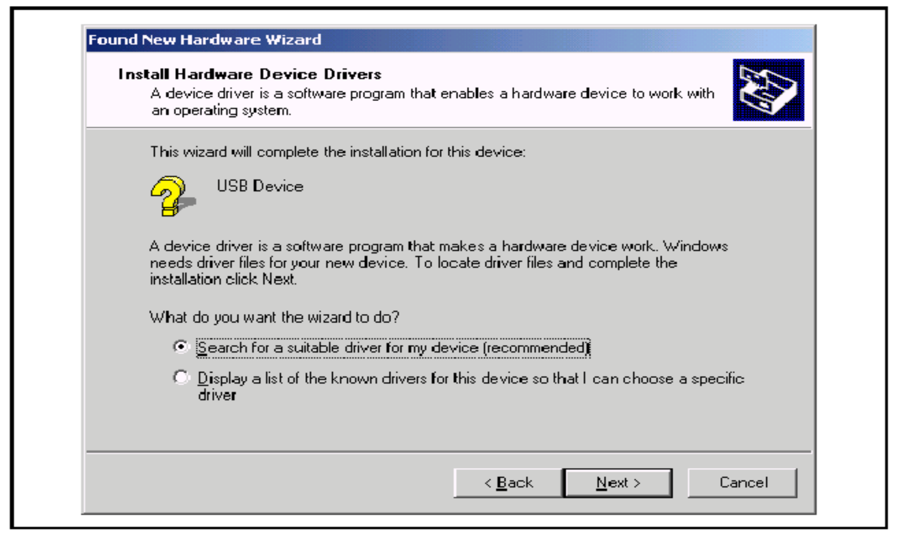
当第一次将一个新的 USB 设备接到 Windows 2000 系统时，一个对话框会出现显示系统发现了一个新的硬件设备. 也可能出现一对话框告诉用户设备数据库正在被建立或更新.

这些对话框出现之后，会显示"发现新硬件向导"对话框. 点击下一步继续安装.

4.3.2 步骤 2

确认"搜索合适的驱动…"被选择,点击下一步.

FIGURE 4-8: WINDOWS 2000 NEW HARDWARE WIZARD - 2

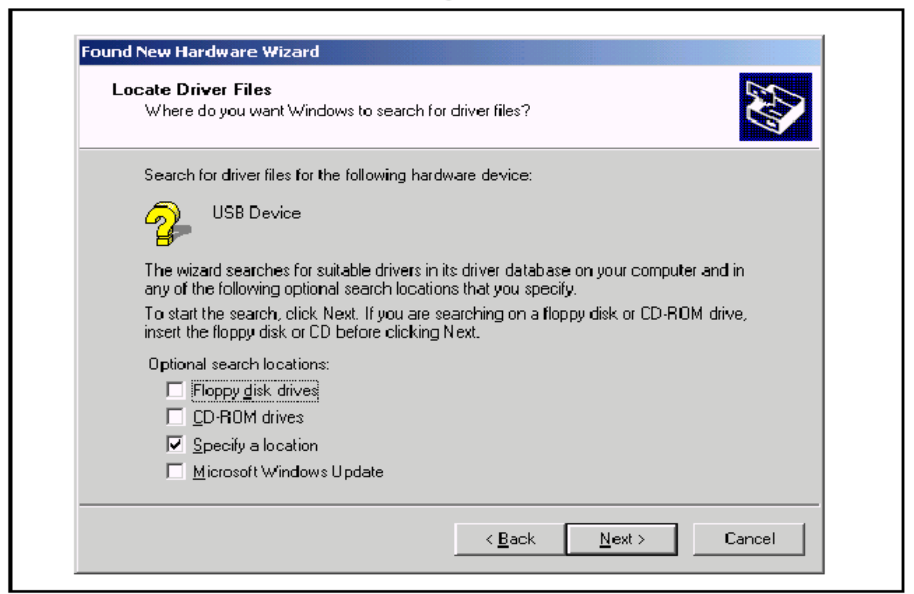


4.3.3 步骤 3

这个对话框允许用户指定哪里可以找到新设备所需要的文件. 确认所有的选框都未选中, 除了提示"指定位置"的那一个. 选择"指定位置", 然后点击下一步.

注意: 如果在下面的步骤中没有发现或识别 INF 或设备驱动文件, 重复步骤 2, 选择"显示已知驱动列表…". 下一屏应该询问设备类型. 这时点击磁盘安装按钮. 接着的对话框允许选择可以找到.INF 文件的路径. 一旦这个路径被选择, 点击 **OK** 继续安装.

FIGURE 4-9: WINDOWS 2000 NEW HARDWARE WIZARD - 3



4.3.4 步骤 4

在标记"拷贝厂商文件"编辑区域输入设备驱动文件的路径. (可以键入路径，或使用浏览按钮进行搜索). 在普通安装下，这个路径应该是:

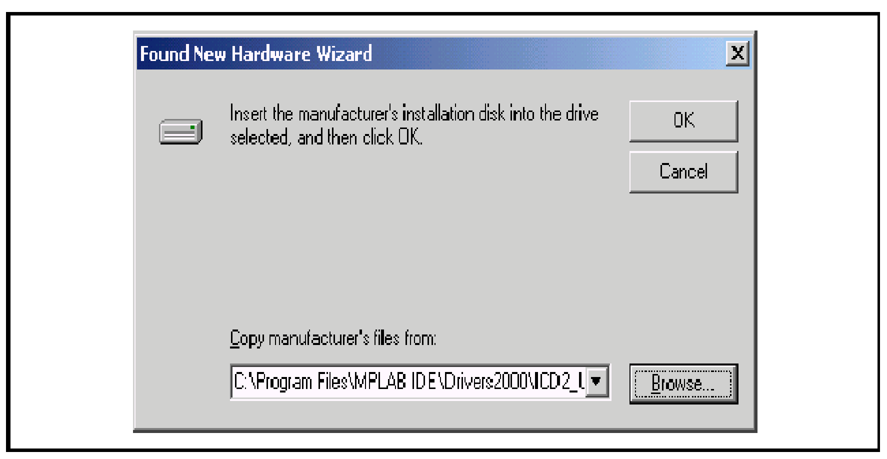
C:\Program Files\MPLAB IDE\Drivers2000\ICD2_USB

这个路径必须包含下面的文件:

- ICD2W2K.INF
- ICD2W2K.SYS
- ICD2W2K L.SYS

一旦路径被指定，点击 **OK.**

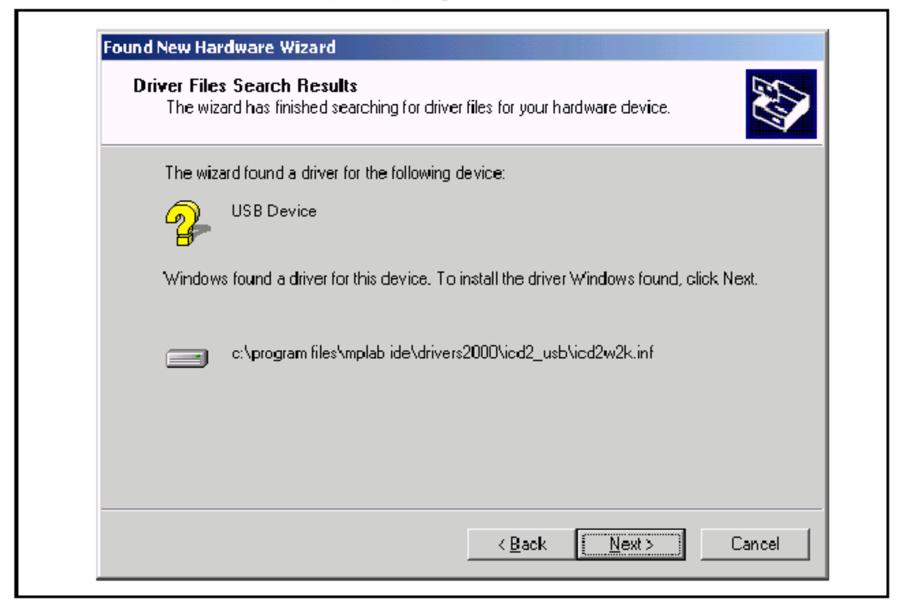
FIGURE 4-10: WINDOWS 2000 NEW HARDWARE WIZARD - 4



4.3.5 步骤 5

下一个对话框应显示向导在上一步指定的位置发现一个设备驱动. 点击下一步继续.

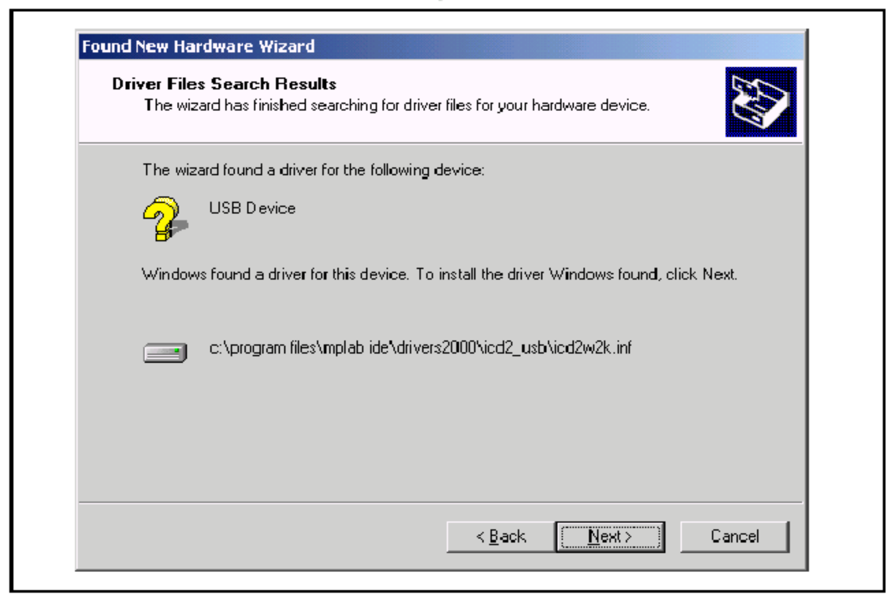
FIGURE 4-11: WINDOWS 2000 NEW HARDWARE WIZARD - 5



4.3.6 步骤 6

下一个对话框应该显示"**Microchip MPLAB ICD Firmware Loader**"被发现. 并显示选择的路径. 点击下一步.

FIGURE 4-12: WINDOWS 2000 NEW HARDWARE WIZARD - 6



最后一个对话框将显示"**Microchip MPLAB ICD2 Firmware Loader**"被安装. 要结束安装过程，点击完成.

这时 MCD2 会再次被检测到. 系统应该自动载入 MCD2 USB 客户端驱动. 在新的对话框请求另一个路径时，请指定与第四步相同的路径.

FIGURE 4-13: WINDOWS 2000 NEW HARDWARE WIZARD - 7



4.4 WINDOWS XP 安装

4.4.1 步骤 1

当第一次将一个新的USB设备接到Windows XP系统时，一个对话框会出现显示系统发现了一个新的硬件设备．也可能出现一对话框告诉用户设备数据库正在被建立或更新.

这些对话框出现之后，会显示”发现新硬件向导”对话框．选择”从列表或指定位置安装(高级)”,点击下一步继续安装.

FIGURE 4-14: WINDOWS XP NEW HARDWARE WIZARD - 1



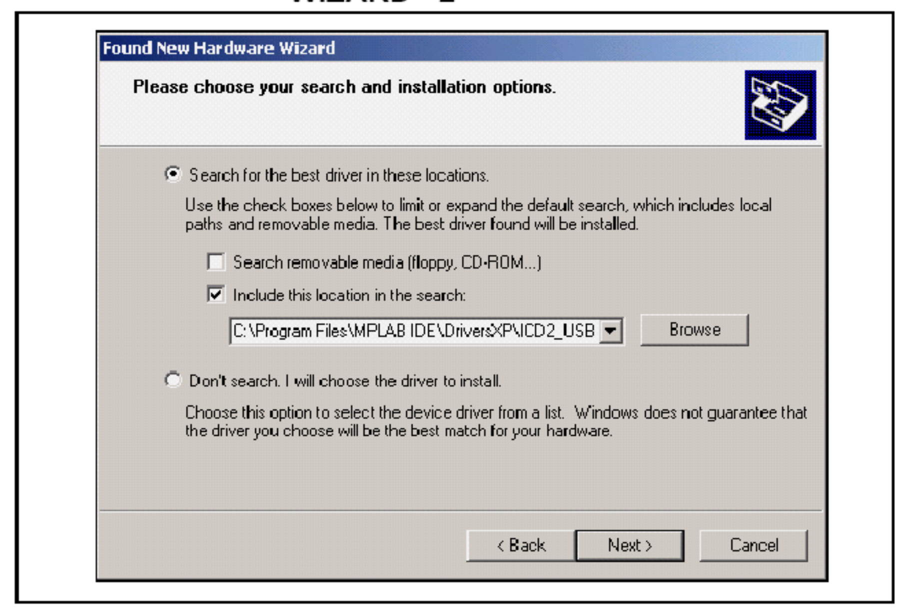
4.4.2 步骤 2

确认"搜索最佳驱动…"被选中. 不要选择"搜索可移除的媒体", 选中"在搜索中包括这个位置". 浏览 MPLAB IDE 安装目录, 选择子目录 **DriversXP\ICD2_USB.**点击下一步.

在安装过程中, 一个对话框会弹出提示" 你为这个硬件安装的软件未通过 Windows 的测试", 点击仍然继续.

注意: 如果在下面的步骤中没有发现或识别 INF 或设备驱动文件, 重复步骤 2, 选择"不要搜索, 我将自己选择…". 然后点击下一步. 下一屏应该询问设备类型. 确认"显示所有设备"被选择, 然后点击下一步. 在"选择设备驱动…"窗口, 点击磁盘安装按钮. 接着的对话框允许选择可以找到.INF 文件的路径. 浏览 MPLAB IDE 安装目录, 选择子目录 DriversXP\ICD2_USB. 一旦这个路径被选择, 点击 **OK** 继续安装.

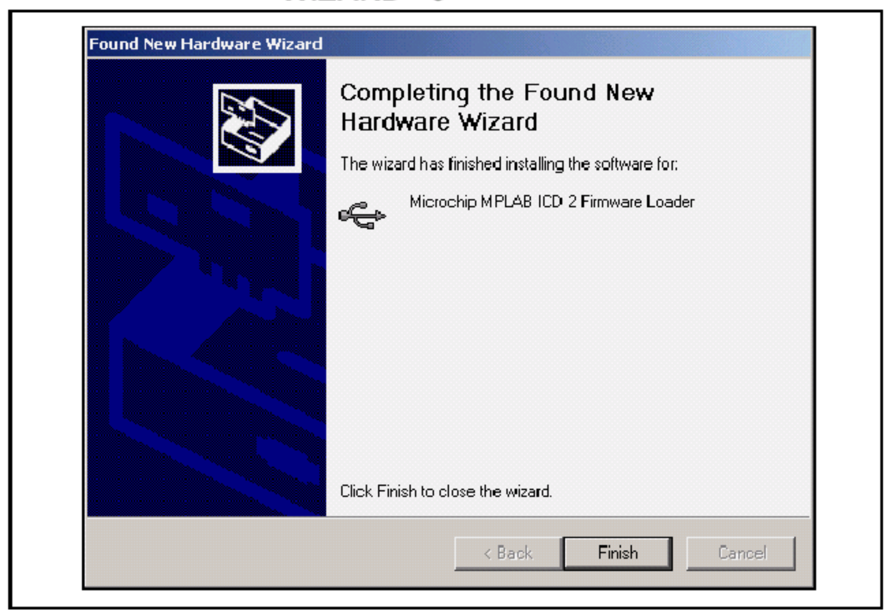
FIGURE 4-15: WINDOWS XP NEW HARDWARE WIZARD - 2



4.4.3 步骤 3

一旦 Microchip MPLAB ICD2 固件载入器被安装, “完成被发现新硬件向导”将会出现. 点击完成结束安装, 并开始固件客户端的安装.

FIGURE 4-16: WINDOWS XP NEW HARDWARE WIZARD - 3



4.4.4 步骤 4

一个对话框将出现显示系统完成添加硬件设备，并立即声明发现了一个新的硬件设备.

在这个对话框出现后，发现新硬件向导对话框会再次显示. 这次将安装 Microchip MPLAB ICD2 固件客户端. 选择"从列表或指定位置(高级)"，然后点击下一步继续安装.

FIGURE 4-17: WINDOWS XP NEW HARDWARE WIZARD - 4

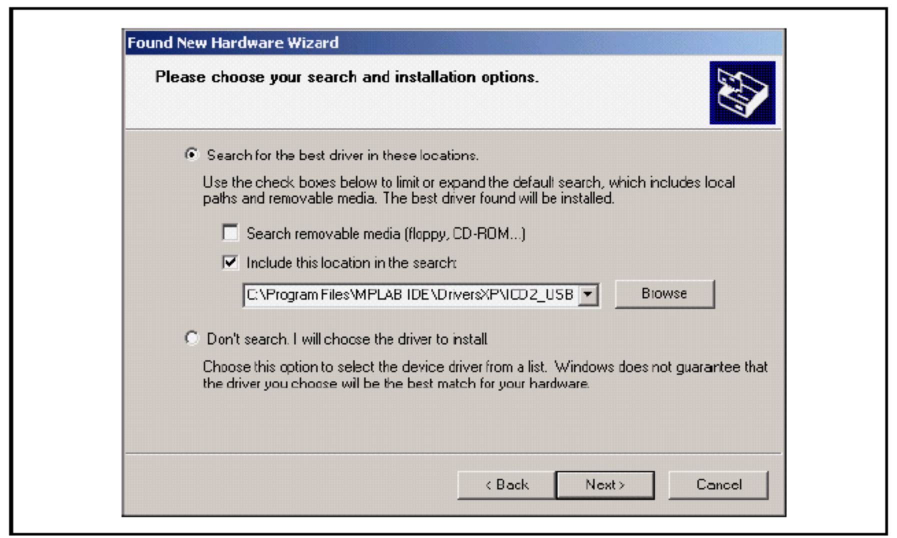


4.4.5 步骤 5

确认"搜索最佳驱动…"被选中. 不要选择"搜索可移除的媒体", 选中"在搜索中包括这个位置". 浏览 MPLAB IDE 安装目录, 选择子目录 DriversXP\ICD2_USB.点击下一步.

在安装过程中, 一个对话框会弹出提示" 你为这个硬件安装的软件未通过 Windows 的测试", 点击仍然继续.

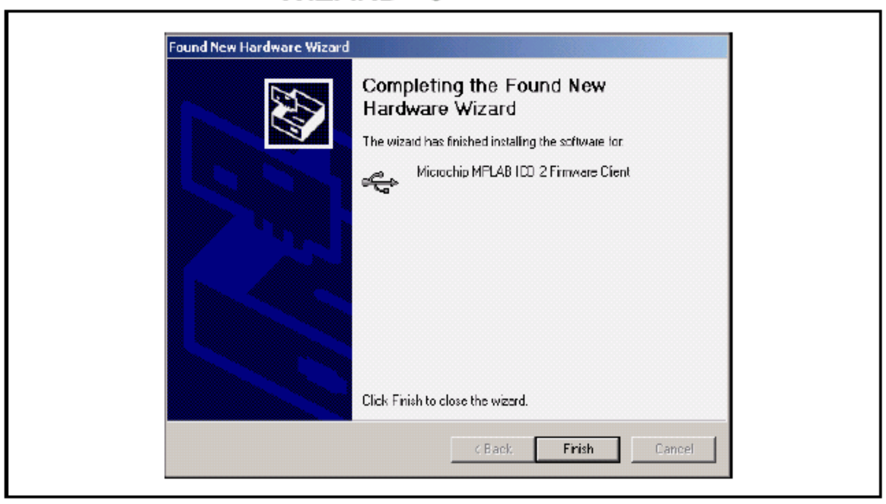
FIGURE 4-18: WINDOWS XP NEW HARDWARE WIZARD - 5



4.4.6 步骤 6

一旦 Microchip MPLAB ICD2 固件客户端被安装, “完成被发现新硬件向导”将会出现. 点击完成结束 USB 安装.

FIGURE 4-19: WINDOWS XP NEW HARDWARE WIZARD - 6



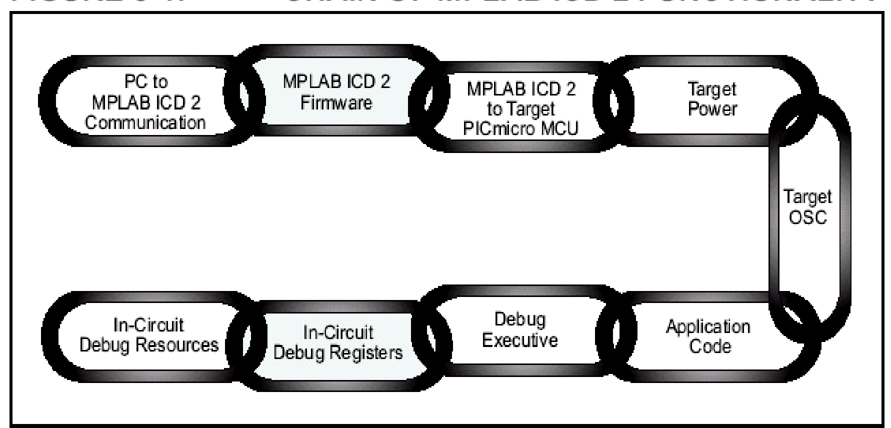
第五章 故障诊断

5.1 出现故障时该如何处理

由于在 MPLAB IDE, MCD2（mplab-icd2）和 PIC 单片机目标应用电路中的很多因素都能影响 MCD2 的正常运行, 所以理解在 MCD2 功能链中的每一环的操作是很重要的.

图 5-1 是显示各个链接的一个框图. 要使 MCD2 编程和调试, 每一个链接都必须工作正常.这一节讨论了这些链接以帮助发现和修复故障.

FIGURE 5-1: CHAIN OF MPLAB ICD 2 FUNCTIONALITY



5.1.1 链接: PC 与 MCD2 的通讯

这一链接是所有子链接的基础. 如果它不工作, 用户将遇到 MPLAB IDE 通讯的错误. 确认 MPLAB USB 驱动被正确的安装. 查看 WINDOWS 硬件管理器对话框的 USB 部分, 看 MCD2 驱动是否正确的安装到 PC 上. 如果使用 RS-232, 确认在 COM 驱动对话框中 FIFO 缓冲被关闭, 并且流量控制设置为硬件方式. 如果 MCD2 USB 驱动在硬件管理器的 USB 部分不可见, 进入到 MPLAB IDE 安装路径, 打开\DriversXXX 文件夹(XXX 对应操作系统, 即 98, NT, XP 等). 使用浏览器打开.HTM 文件, 查看相应操作系统的驱动安装细节. 更多 USB 安装细节请查看第四章.

5.1.2 链接: MCD2 固件

确认在 MCD2(MPLAB-ICD2)中载入了最新版的固件. 更新的固件可以通过 MPLAB IDE 软件来下载. 最新版的软件应该总能从 Microchip 网站下载到.

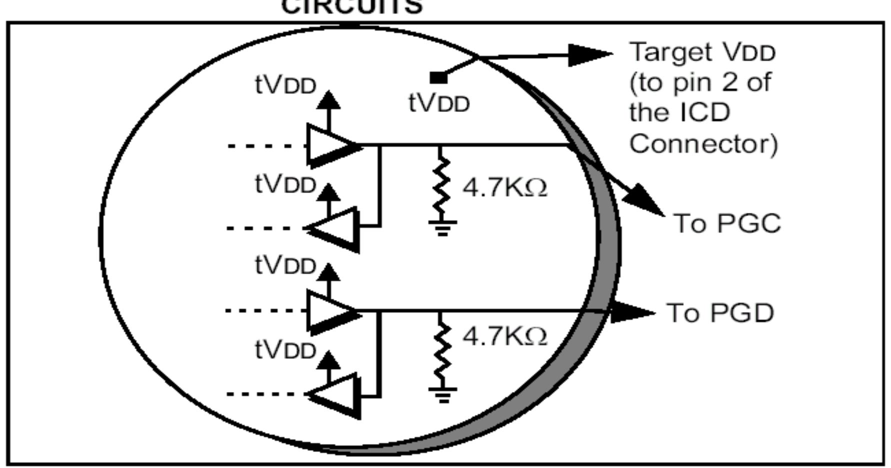
固件的版本将决定哪些 PIC 单片机被支持. 关于固件和 MCD2(MPLAB-ICD2)的最新信息,请查看 README 文件.

5.1.3 链接: MCD2 连接到目标 PIC 单片机

确认五根线都正确的连接, 并且在 VPP, PGC 和 PGD 上没有其它信号或元件对 MCD2 信号有干扰. 如果 MCD2 的 VDD 和 VSS 没有连接到目标电源和地, 它将不能工作. 当 MCD2 提供电源给目标电路时, 它能提供

VDD 到目标(<200mA，仅限 5V)和这些输出缓冲．否则，目标的 VDD 将被使用．VDD 被 MCD2 感应，以检测电压值是否正确．从目标 VDD 供电允许电压等级转换到目标低电压运行．对于 MCD2(MPLAB ICD2)内部缓冲电路的简化电路如图 5-2 所示．

FIGURE 5-2: MPLAB ICD 2 INTERNAL BUFFER CIRCUITS



如果 MCD2 在它的 VDD 线上没有电压，它将不能运行．当 PGC 和 PGD 通讯时，测试它们管脚的信号，可以看到目标从 VDD 到 VSS 的波形变化．依据被执行的操作, VPP 应显示+12,+5 和 0V．只有在从 MPLAB 用户接口执行 MCD2 功能之后，在哪些线上才能看到信号．否则，这些连线应该为空闲．

5.1.4 链接：目标电源

MCD2 能在目标 PIC 单片机的 VDD 约为 2V 至 5.5V 的范围内工作．应用板必须通过它自己的电源供电．记住，MCD2 上的 PGC 和 PGD I/O 驱动器是从目标 VDD 来供电的．

5.1.5 链接：目标振荡器

当编程一个器件时，它将不需要一个运行的振荡器来编程．但调试功能将不能工作，直到目标振荡器正常工作．不像在线仿真器，当调试时，MCD2 要求目标 PIC 单片机运行．

要确认目标系统正确的执行指令，测试是必要的．例如，编一个小程

序让 LED 闪烁，然后关闭 MCD2 的调试功能，将程序编程到目标 PIC 单片机.

断开 MCD2 并复位目标. 如果 LED 不闪烁，检查电路以发现它为什么不工作. 如果在调试功能被关闭的情况下对目标器件编完程，MCD2 仍然被连接着，MPLAB 能控制住复位. 如果它独立的运行但没与 MCD2 连接(在选择 ***Debugger>Run*** 之后)，那么互连可能有问题. 使用示波器查看 PGC，PGD 和 VPP. 确认振荡器启动迅速. 如果振荡器启动时间过长，MCD2 将超时，并给出错误.

确认正确的振荡器模式被选择(查看 ***Configure>Configurations Bits***).振荡器模式将依赖振荡器的选择. 对于一个晶体尝试 HS 模式，对于外部阻/容振荡器，尝试 RC 模式. 这些设置对于最终设计可能需要改变，但它们是最容易上手的一些方式. 在配置位被设置,并在调试或普通模式下加了电源后，振荡器的运行可以通过 OSC2 来检查.

5.1.6 链接：应用代码

要校验 MCD2 正确的编程，可以运行 LED 流水灯的测试程序，或编程代码到目标器件中. 在编程之后 MCD2 将执行一个校验功能，以确认目标 PIC 单片机中的数据与 MPLAB 的程序存储器相匹配. 也可以选择 ***Programmer>Verify*** 来手动的比较目标 PIC 存储器与 MPLAB IDE 的程序. 如果目标 PIC 单片机有 AVDD 和 AVSS，确信它们被正确的连接. 这些连线应相应的连接到电源(VDD)和地(VSS). 更多细节请参考器件数据手册. 如果存在这些引脚而没有被连接的话，MCD2 将不能工作. 确认低电压编程被关闭(***Config>Configuration Bits…***).

5.1.7 链接：调试执行代码

当从 MPLAB IDE 使用 ***Debugger>Program*** 功能，调试执行代码将被下载，在目标 PIC 单片机中的在线调试寄存器将被使能. 当进行调试时，编程将总是从调试菜单来选择，而不是编程器菜单. 当从编程器菜单来编程(在关闭 MCD2 作为调试器)，调试执行代码将不被下载，在线调试寄存器也将不被使能.

通过查看 ***Config>Configuration Bits…***菜单选择，它能决定 MPLAB 是否将要下载调试执行代码. 如果位"**Background Debug**"被使能，那么当器件编程时调试执行代码将被下载.

5.1.8 链接：在线调试寄存器

确认在编程之前，在 MPLAB 中的 MCD2 上调试功能被使能．通过查看 ***Config>Configuration Bits…***对话框，它能校验在线调试寄存器被使能．标记”**Background Debug**”的行应显示”Enabled”．编程应该从 ***Debug*** 菜单进行，而不是 ***Programmer*** 菜单．

5.1.9 链接：在线调试资源

在在线帮助中查看针对某一器件使用的特定的寄存器．被 MCD2 保留的文件寄存器或程序存储器不能使用．如果使用 xxxxxi.lkr 链接描述文件(名字以’i’结束)，那么，除非描述文件被修改，这些资源将被标记为保留，并且将不能在应用中使用．如果链接器没有被使用，在代码中使用 **CBLOCK** 或 **EQU** 用于变量的存储，确认被 MCD2 要求的寄存器没被使用．如果代码保护或表读保护被使能，如果看门狗定时器在运行，或如果振荡器没有通过配置位设置为正确的模式，在线调试器将不能工作．如果使用快速中断或 **CALL FAST** 指令，MCD2 使用了影子堆栈，用户将不能从快速中断程序或 **CALL FAST** 功能正确的退出．

第六章 常见问题

这一章谈到的一些常见问题:

6.1.1 为什么我的系统编程或校验失败?
6.1.2 为什么我需要 ICD 头适配器?
6.1.3 我不能连接到 MCD2．现在我该怎么做?
6.1.4 MCD2 应答”Target not in debug mode error.”，这表示什么意思?
6.1.5 MCD2 能将目标器件在低电压下运行吗?
6.1.6 MCD2 支持低电压编程吗(LVP)?
6.1.7 当我配置 PLL 振荡器时，为什么会出现问题? MCD2 被挂起．
6.1.8 当我安装驱动时，为什么驱动无法找到，尽管我能在驱动文件夹中看到它们，并且我将驱动向导指向正确的文件夹?
6.1.9 我能使用 MCD2 做代码保护吗?
6.1.10 MCD2 是如何处理校验数的?
6.1.11 为什么我从我的 EEDATA 区域获得错误的值?

6.1.1 问：为什么我的系统编程或校验失败?

答：按照第五章所描述的检测 PGC，PGD 和 VPP 连接．确认目标 PIC 单片机有电源提供．如果目标 PIC 单片机有 AVSS 和 AVPP 引脚，校验这些引脚都正确的连接．

6.1.2 问：为什么我需要仿真头适配器?

答：对于 MCD2 支持的低引脚数的器件，如果在线调试引脚被保留，它们将不能有效的被使用．试想一下，对于 8 脚的器件, 6 个 I/O 将失去其中的 3 个，这是不能被接受的．针对这个原因，生产了特殊的 PIC 单片机通过一个适配头来仿真这些低引脚数的器件．这将允许在目标应用中使用所有的引脚．这些特殊的 PIC 单片机具有在线通讯引脚与 MCD2 接口．

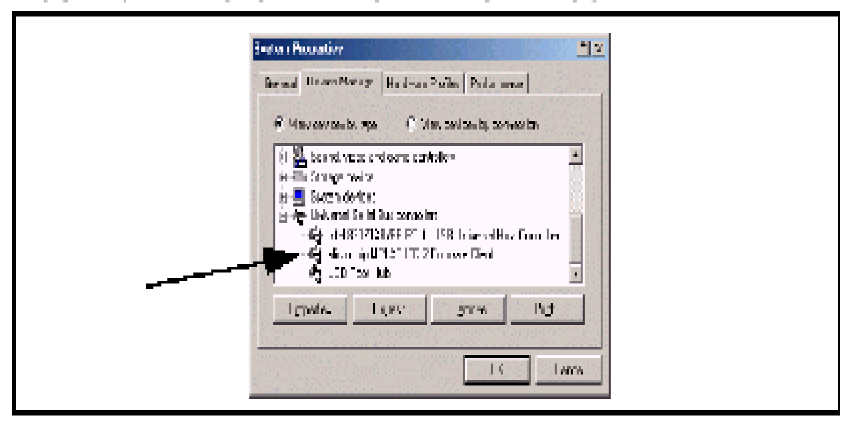
这样做的优点是可以在 MCD2 上使用低引脚数的器件来开发．缺点是这些器件不能直接与MCD2连接使用．这些定制的PIC单片机类似于仿真器芯片．它们能支持不止一个型号．在适配头上有跳线用来配置芯片以匹配相应的开发器件．这些低引脚数的器件能使用 MCD2 的在线下载来编程，方法是在目标应用板上放置一个 ICD2 连接座，将这些器件的 VPP，PGC

和 PGD 连接.

6.1.3 问: 我不能连接到 MCD2. 现在我该怎么做?

答: MCD2 的电源灯亮了吗? 发光管应该被点亮. 如果它比较暗淡, 可能是只连接了 USB, 用户可能需要再接一个电源. RS-232 连接不能给 MCD2 供电, 注意,有些 USB 不能提供电源. 针对当前的故障线索查看在线帮助. USB 的驱动安装是否正确? MCD2 USB 驱动应该在 WINDOWS 设备管理器对话框中应该可以看到. 一些 USB 集中器不能为相应的 USB 设备提供电源. 那么使用这些集中器时, 需要一个电源与 MCD2 连接.

FIGURE 6-1: SYSTEM PROPERTIES DIALOG



6.1.4 问: MCD2 应答"Target not in debug mode error.", 这表示什么意思?

答: 通常这意味着 MCD2 不能与调试执行程序通讯. 要将调试执行程序下载到目标 PIC 单片机, 只能通过 ***Debugger>Program*** 菜单的选择来编程器件.也有其它的原因导致调试执行程序不能通讯, 如目标时钟或电源的问题. 检查配置位, 看"**Background Debug**"是否被使能. 查看 ***Config>Configuration Bits…***, 确认看门狗定时器被关闭, 代码保护被关闭, 振荡器设置正确.

6.1.5 问: MCD2 能将目标器件在低电压下运行吗?

答: 能. 只要目标 PIC 单片机支持低电压运行, 它就能运行在约 2V 的低电压下. 在 MCD2 的输入/输出缓冲器中, 有电压级别转化器. 它们从目标器

件的 VDD 来供电. 并且, MCD2 将感应目标器件的操作电压, 并正确的调整它的功能以处理这一操作, 即使用正确的 FLASH 擦除算法. 在 MCD2"**Settings**"对话框中, VDD 需要选择"From Target", 并且在目标板上要有一个电源用于低电压运行.

6.1.6 问: MCD2 支持低电压编程吗(LVP)?

答: 不能. 但这并不意味着当目标器件运行在低电压 VDD 时它不能正常工作. 它只是表示施加到 VPP 的编程电压总是+13V.

6.1.7 问: 当我配置 PLL 振荡器时, 为什么会出现问题? MCD2 被挂起.

答: 这是 PIC 单片机的一个要求. 在将配置位编程到 PLL 振荡器, 电源需要被断开然后再加到目标板上. 如果没有这样做, 目标 PIC 单片机将不会有时钟. 没有时钟, 调试模式将不能工作. 同样, 如果在切换到 PLL 模式时电源没有被断开和再连接, 器件可能会运行, 但没有使用 PLL.

6.1.8 问: 当我安装驱动时, 为什么驱动无法找到, 尽管我能在驱动文件夹中看到它们, 并且我将驱动向导指向正确的文件夹?

答: 碰到这个问题, 可以退出驱动安装向导, 进入控制面板的"添加新硬件". 在系统搜索新硬件后, 选择"不, 设备不在列表中." 然后选择"不, 我想从列表中选择硬件". 然后选择" 通用串行总线控制器", 当"磁盘安装"按钮出现时, 进入到驱动文件夹, 选择正确的驱动.

6.1.9 问: 我能使用 MCD2 做代码保护吗?

答: 不能. 代码保护, 特别是在程序存储器中任何区域的表读保护都将阻止 MCD2 正常的工作. 当使用 MCD2 进行调试时, 不要使用任何的代码保护或表读保护的配置设置. 当编程一个器件用于测试时, 可以使能代码保护.

6.1.10 问: MCD2 是如何处理校验数的?

答: 它是自动处理的. 在 MCD2 进行擦除, 编程和调试操作时, 在程序存储器中被 PIC 单片机使用的用于校验数的任何值都将被读出并保存. 所以不需要任何操作来保护这一数据.

6.1.11 问: 为什么我从我的 EEDATA 区域获得错误的值?

答: MCD2 能直接读取 EEDATA 区域, 而不需要进行 EECON 寄存器要求的 TABLRD 指令顺序. MPLAB 使用的缓冲器有时会干涉用户的代码. 当单步执行代码时, 应避免来回切换从程序读数据到 MCD2 读 EEDATA 区

域.

6.1.12 问: 为什么"Erase All Before Programming"变成灰色?

答: 在一些较新的 FLASH 器件中, 编程算法要求非邻近的程序存储器区域按 bank 被编程. 对于这些器件, 在编程之前所有的存储器必须被擦除.

6.1.13 问: 如果不与 MCD2 连接, 我的程序能从 PORTB 或 GPIO 进行读和写吗?

答: 能. 当在线调试功能被使能时, PGC 和 PGD 总是被 MCD2 使用, 从 PORTB 对用户代码进行读或些将不会受到干扰. 注意, 从 PGC 和 PGD 读的值不一定是正确的, 对这两个引脚的写操作将被忽略. 另外, 如果 PORTB 电平变化中断被使能, PGC 和 PGD 上的信号将不能产生中断.

6.1.14 问: 当单步运行时, 为什么定时器运行不正常?

答: 这是使用在线调试器的一个缺点. 由于代码实际是在调试执行代码中运行, 在调试执行代码运行期间, 定时器会持续的运行, 即使用户的应用的程序被中止.

6.1.15 问:当我使用 PIC12F629/675 或 PIC16F630/676 时, 为什么会有警告和错误?

答: 当使用 MCD2 时, 这些器件的 GP1/RA1 引脚不能被拉高. 关于针对这些器件使用 MCD2 的更多信息, 请参考 microchip 的 MPLAB ICD2 适配头文档(DS51292).

6.1.16 问: 什么因素会使电源和忙 LED 闪烁?

答: 这可能表示目标MCD2连接头的接线顺序不对(与接线框图方向相反). 闪烁表示 MCD2 由于高电流被关断. 下面的测试表示目标被反序连接:

• 查看闪烁的电源和忙LEDs(电源可能一起断掉).

• 执行一遍"**Self Test**", 查看MCLR=VPP上的错误(其它所有的测试可能都会通过).

• 使用目标控制器的默认地址范围, 执行一次编程. 在引脚1(VPP)上7-8V 的电压等级太低.

注意, 如果目标被反序连接, MCD2中的保护电路将阻止对模块的破坏.

当目标器件连线正确, 应该看到正常的操作.

6.1.17 问: MCD2自检做些什么?

答: 在确定 MCD2 模块或目标连接的问题时, 自检是很有用的.

Target VDD

如果选择了"**Power from ICD2**"，将测试从 MCD2(只能为 5V)提供的 VDD.

如果没有选择"**Power from target**"，将测试从目标板(2 到 6V)提供的 VDD.

通过/失败代码:

00=通过　　VDD 在指定范围之内
01=最小错误　　VDD 低于指定范围
80=最大错误　　VDD 高于指定范围

错误的出现表示不正确的电源设置

模块 VPP

在编程过程中，测试从 MCD2 提供给目标 VPP/MCLR 引脚的编程电压(VPP).

通过/失败代码:

00=通过　　VPP 在指定范围之内
01=最小错误　　VPP 低于指定范围
80=最大错误　　VPP 高于指定范围

出现错误表示目标 VPP/MCLR 引脚接线不正确.

MCLR=GND

测试 MCD2 提供复位电压给目标 VPP/MCLR 引脚用于目标复位的能力.

通过/失败代码:

00=通过　　地能提供给目标 VPP/MCLR 引脚
80=最大错误　　地电压对于 VPP/MCLR 引脚太高

出现错误表示目标 VPP/MCLR 引脚接线不正确.

MCLR=VDD

测试在正常操作过程中(如'Run')MCD2 提供 VDD 给目标 VPP/MCLR 引脚的能力.

通过/失败代码:

00=通过　　VDD 能提供给目标 VPP/MCLR 引脚
01=最小错误　　VDD 对于目标 VPP/MCLR 引脚太低
80=最大错误　　VDD 对于目标 VPP/MCLR 引脚太高

出现错误表示错误的电源设置.

MCLR=VPP

测试在编程操作过程中 MCD2 提供 VPP 给目标 VPP/MCLR 引脚的能力.

通过/失败代码:

00=通过 VPP 能提供给目标 VPP/MCLR 引脚

01=最小错误 VPP 对于目标 VPP/MCLR 引脚太低

80=最大错误 VPP 对于目标 VPP/MCLR 引脚太高

出现错误表示目标 VPP/MCLR 引脚连线不正确.

6.1.18 问: 当在 RETFIE 指令使用高优先级中断时，为什么 W,STATUS 和 BSR 寄存器会改变?

答: CALL FAST 和用于高优先级中断的影子寄存器被 MCD2 使用了. 这些对于 MCD2 操作是保留资源. 如果断点设置在 CALL FAST 子程序内，或在针对高优先级中断的服务程序中，通过 RETURN FAST 或 RETFIE 指令使用了影子寄存器，那么将会遇到问题.

6.1.19 问: 当我在我的程序的起始位置设置一个断点时，为什么它停止在地址 0001，而不是 0000?

答: MCD2 停止在断点之后的指令. 这表示在地址 0000 的断点会被执行，然后当它获得一个断点时程序计数器将指向地址 0001. 如果用户需要在他们代码的第一条指令处停止，他们必须在地址 0000 放置一条 NOP.

6.1.20 问: 为什么我的校验存储器显示为擦除值?

答: MPLAB IDE 显示的是默认的存储器值. 要显示这个器件真实的值，必须使用 MCD2 进行一次器件读操作.

6.1.21 问: 当单步运行代码时，我的定时器计时溢出，但为什么我的定时器中断程序没有执行?

答: 当单步运行时，在线调试器将不允许 PIC 单片机响应中断. 如果允许的话，并且用户有外部中断，那么单步运行将总是在中断程序结束. 要调试一个中断，可以在中断服务程序内设置一个断点并运行，这样在发生中断后进入断点.